Generative Adversarial Nets

Machine Learning

Sebastian Bek

University Heidelberg
Seminar "Biomedical Image Analysis: Deep Learning"
Supervisors: Karl Rohr & Christian Kromm
sebibek@gmail.com

17. Juli 2019

Outline

Outline

Motivation

- **Easy-to-Train**: propose a new framework for estimating generative models, with the ability to be trained by a standard backpropagation algorithm, through an underlying adversarial process
- **Requirements**: quality (proximity to prior distribution) and quantity (diversity)
- **Disuse of complicating training and sampling algorithms**: e.g. MCMCs (Markov Chain Monte Carlo Methods), approximate inference, score matching, NCE (noise-contrastive estimation)

Related Work

Tab.: Challenges in generative modeling:

	Deep directed graphical models	Deep undirected graphical models	Generative autoencoders	Adversarial models
Training (Issues)	Inference needed	Inference needed, MCMC needed to approximate partition function gradient	Tradeoff between mixing and power of reconstruction generation	Synchronizing the discriminator with the generator, Helvetica scenario
Sampling	No difficulties	Requires Markov chain	Requires Markov chain	No difficulties
Evaluating $p_g(x)$	may be approximated with AIS	may be approximated with AIS	may be approximated with Parzen density estimation	may be approximated with Parzen density estimation
Model design	Nearly all models incur extreme difficulty	Careful design needed to ensure multiple properties	Any differentiable function is theoretically permitted	Any differentiable function is theoretically permitted

Analogy: Counterfeiters and Police

Police and Counterfeiters

Generator G: Counterfeiters, producing and using fake curreny

Discriminator D: Police, trying to detect the counterfeit currency and to discern it from the genuine one

Fig.: Indicators of a genuine dollar bill, *Source:* ①

Outline

1. Introduction
2. Adversarial Nets
3. Theory
4. Evaluation
5. Summary

Fundamentals

Important Terms

$$\begin{aligned} \text{prior (data) distribution} &: p_{data} = p_z(z) \sim z \\ \text{generator function} &: G(z, \theta_g) \qquad \theta_g : \text{generator parameters} \\ \text{discriminator function} &: D(x, \theta_d) \qquad \theta_d : \text{discriminator parameters} \end{aligned}$$

whereas $D(x)$ represents the propability that x is drawn from the training samples (prior data distribution) rather than p_g.

Mathematical Formulation

Formal Definition

Two-Player Minimax (specific: zero-sum) game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

$$\Rightarrow \max V(D, G) = 0 \qquad \min V(D, G) = -\infty$$

outer loop: minimize $V(D, G)$ for 1 step | inner loop: maximize $V(D, G)$ for k steps

$\rightarrow$ prevents the network from overfitting the training data

$\rightarrow$ stop when gradient of G is very small $\Rightarrow$ slowly changing ($\ddot{G} \approx 0$)

Precautions

Formal Definition

Two-Player Minimax (specific: zero-sum) game with value function $V(G, D)$:

$$\min_D \max_G V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z))]$$

$$\Rightarrow \max V(D, G) = 0 \qquad \min V(D, G) = -\infty$$

Caveat: Early Learning

$\rightarrow$ Gradients of G might be poor/vanishing:

$\Rightarrow$ D can reject samples with high confidence ($\log(1 - D(G(z))$ saturates in this case)

$\Rightarrow$ instead of minimizing $\min \log(1 - D(G(z))$, we can assign G to maximize $\log D(G(z)$ $\rightarrow$ same fixed point of dynamics, but yields much stronger gradients

SGD: Stochastic Gradient Descent

Gradients

Discriminator Gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^{(i)}) + \log(1 - D(x^{(i)}))]$$

Generator Gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [\log(1 - D(x^{(i)}))]$$

Outline

1. Introduction
2. Adversarial Nets
3. Theory
4. Evaluation
5. Summary

Architecture

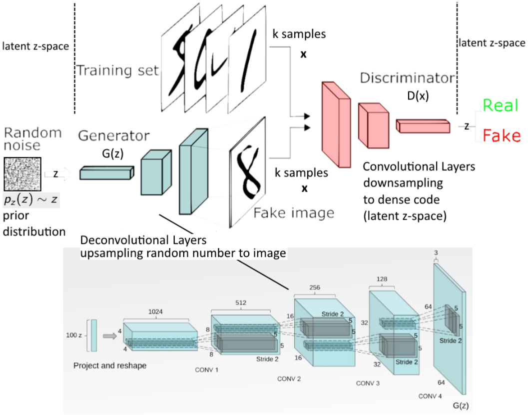


Fig.: Visualization of Architecture (general/DCGAN), *Source:* 2

Global Optimum

Global Optima

Discriminator Optimum:

$$D_G^*(\mathbf{x}) = \frac{p_{data}(\mathbf{x})}{p_{data}(\mathbf{x}) + p_g(\mathbf{x})}$$

Generator Optimum:

$$p_g = p_{data} = p_z \Rightarrow D_G^*(\mathbf{x}) = \frac{p_{data}(\mathbf{x})}{p_{data}(\mathbf{x}) + p_g(\mathbf{x})} = \frac{1}{2}$$

Convergence of the Algorithm

Adversarial Nets

Adversarial Nets represent a limited family of generative p_g distributions via the function $G(z, \theta_g)$ with p_g being indirectly optimized through optimization of θ_g. Lack of theoretical validity of multilayer perceptrons, but excellent performance in practice.

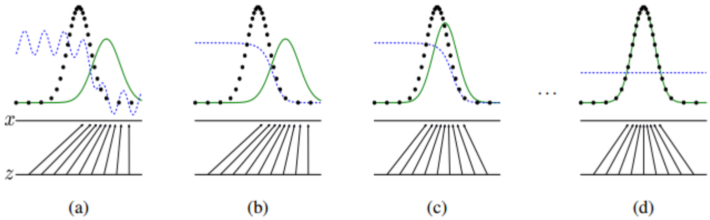


Fig.: Visualization of distributions/mapping $z \rightarrow x$, *Source:* 3

Outline

Training

Datasets

- MNIST
- TFD: toronto face database
- CIFAR-10 [21]

Experiments

Test scenario: Estimation of propability of "test set data" under (behind) $p_g \rightarrow$ fit a Gaussian Parzen window to the samples generated with G and report the log-likelihood considering test data samples obtaining variance σ from cross-validation with the test dataset.

Model	MNIST	TFD
DBN [3]	138 ± 2	1909 ± 66
Stacked CAE [3]	121 ± 1.6	2110 ± 50
Deep GSN [3]	214 ± 1.1	1890 ± 29
GAN	225 ± 2	2057 ± 26

Tab.: Log-likelihood estimates of p_g

Sampling from the Model - 1

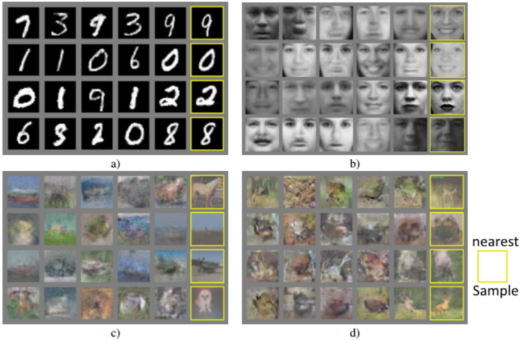


Fig.: Visualization of Samples from the model, *Source:* 3

Sampling from the Model - 2

MNIST dataset

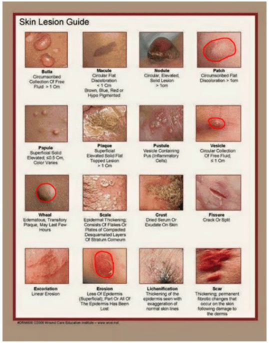
MNIST digits linearly interpolated to evaluate diversity of generative distribution p_g of GAN approach..

Fig.: Linearly interpolating digits in z-space of G after training, *Source:* ③

Application in Biomedical Image Analysis

Application Example: Skin Lesion Segmentation

- Generator G: Fully (de-)convolutional neural network to synthesize (generate) valid segmentation masks
- Discriminator D: another convolutional neural network to distinguish synthetic (fake) from real masks



Skin lesions 101

Fig.: Segmented Skin Lesions, *Source:* [1]

[1] https://suprememedical.com/Product/skin-lesion-guide

Benefits and Drawbacks

Advantages and Disadvantages

⊕ **Advantages**	⊖ **Disadvantages**
• no need for Markov Chains (MCMC-approaches)	• no explicit representation of $p_g(x)$
• training through standard backpropagation	• D needs to be sufficiently synchronized with G (D must stay inline) $\rightarrow$ avoidance of Helvetica scenario:
• incorporation of wide variety of functions possible $\rightarrow$ applicability	G collapsing too many latent z's (e.g. random variables) to the same sample x, leading to insufficient generative diversity
• able to represent sharp, degenerate distributions	

Tab.: Advantages and Disadvantages of GANs

Outline

Summary

- GANs: new framework for estimating generative models defined by multilayer perceptrons (Convolutional Layers) trained by standard backpropagation $\rightarrow$ no need for any Markov chains (MCMC-approaches), which can have problems Mixing (Converging)
- Results Samples considered to be competitive with those of the state-of-the-art generative models
- Empirical Evaluation (Fitting a Gaussian Parzen window to measure distribution similarity of p_g as log-likelihood metric) indicates comparable scores than achieved for state-of-the-art methods like DBNs, Stacked CAE, GSN, Adversarial nets and comparable validity/representativity

Appraisal

- approach is, besides DGMs (directed graphical models), the only one, inducing no problems or further elaborations for sampling (generating samples) or training $\rightarrow$ rather simple implementation
- experiments show comparable similarities against state-of-the-art methods in log-likelihood score and variance in matching the prior distribution p_z with the generative one (p_g), but the method proposed is regarded to be more simple than others
- the synchronization of D is yet an effortful factor, since sufficient reasoning for the number of steps of the inner loop is needed to avoid the previously named "Helvetica scenario"!
- future applications might involve the synthetic generation of morphologically correct segmentation masks of seperatable objects, fake images (databases), keys/passwords (cryptography), image processing

Outlook

Straighforward extensions:

1. Conditional generative Model $p(\mathbf{x}|c)$ w. adding condition c
2. Learned approximate inference: Predict prior-z w. given latent x
3. Modeling of multiple Conditionals: $p(\mathbf{x}_S|\mathbf{x}_{\not S})$
4. Semi-Supervised Learning: Better Performance w. partially labeled training data
5. Efficiency improvements: Better methods to coordinate G and D, better distributions to sample z from during training

Questions?

Questions?

Sources and further reading

Literature

1. Goodfellow, Ian, et al. "Generative adversarial nets."
2. Izadi, Saeed & Mirikharaji, Zahra & Kawahara, Jeremy & Hamarneh, Ghassan. Generative adversarial networks to segment skin lesions.

Images

1. https://www.sevendaysvt.com/vermont/some-counterfeiters-still-do-it-old-school/Content?oid=3276910
2. https://www.altoros.com/blog/the-diversity-of-tensorflow-wrappers-gpus-generative-adversarial-networks-etc/ & https://medium.freecodecamp.org/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394
3. Goodfellow, Ian, et al. "Generative adversarial nets."